

PENGEMBANGAN *SHORTEST PATH ALGORITHM* (SPA) DALAM RANGKA PENCARIAN LINTASAN TERPENDEK PADA GRAF BERSAMBUNG BERARAH BERUNTAI

Oliver Samuel Simanjuntak

Jurusan Teknik Informatika UPN "Veteran" Yogyakarta
Jl. Babarsari 2 Tambakbayan 55281 Telp (0274) 485323
e-mail : oliver.simanjuntak@upnyk.ac.id

Abstrak

Secara manual, pencarian jarak dan lintasan terpendek pada graf yang memiliki jumlah verteks, arc dan untai yang sedikit mudah dilakukan. Namun bila graf tersebut memiliki verteks, arc dan untai yang besar, pencarian jarak dan lintasan terpendek akan menjadi sukar. Penelitian berhasil mengembangkan Shortest Path Algorithm (SPA) dalam rangka pencarian jarak dan lintasan terpendek pada graf bersambung berarah beruntai. Aplikasi pengembangan SPA memungkinkan pencarian jarak dan lintasan terpendek pada graf bersambung berarah secara cepat dan tepat. Pengembangan SPA menjadi solusi dalam membantu memecahkan masalah pencarian lintasan terpendek pada sebuah graf bersambung berarah beruntai.

Kata Kunci: SPA, pencarian jarak terpendek, lintasan terpendek, graf bersambung berarah beruntai.

1. PENDAHULUAN

Pencarian lintasan terpendek memberi manfaat dalam penghematan waktu, tenaga maupun biaya. Pada kehidupan nyata, distribusi barang atau perjalanan pak pos dari suatu tempat ke tempat lain merupakan contoh-contoh pencarian lintasan terpendek. Pencarian lintasan terpendek dapat disajikan dalam bentuk graf bersambung berarah beruntai. Graf bersambung berarah beruntai adalah himpunan tempat/obyek dan rute tersambung yang memiliki arah dan untai.

Shortest Path Algorithm (SPA) merupakan metode pencarian lintasan terpendek pada sebuah graf bersambung, berarah dan beruntai. Untuk pencarian lintasan terpendek dari sebuah graf bersambung dan berarah, dapat digunakan perhitungan cara manual atau dengan melalui komputer. Pencarian lintasan terpendek pada graf bersambung berarah yang memiliki jumlah verteks dan arc sedikit melalui cara manual akan mudah. Namun bila graf memiliki verteks dan arc berjumlah besar, pencarian lintasan menjadi sukar. Pencarian lintasan secara mudah dan akurat dapat dilakukan dengan bantuan komputer. Dalam pencarian lintasan terpendek, komputer mengandalkan logika yang kita tuangkan dalam sebuah algoritma pemrograman pencariannya.

Masalah yang diangkat dari penelitian ini adalah pencarian lintasan terpendek pada sebuah graf bersambung berarah beruntai. Untuk itu, akan dibuat sebuah pengembangan metode SPA dalam rangka pencarian lintasan beruntai. Aplikasi pengembangan SPA mampu mencari jarak dan lintasan terpendek pada graf bersambung berarah beruntai.

2. TINJAUAN PUSTAKA

2.1 Graf

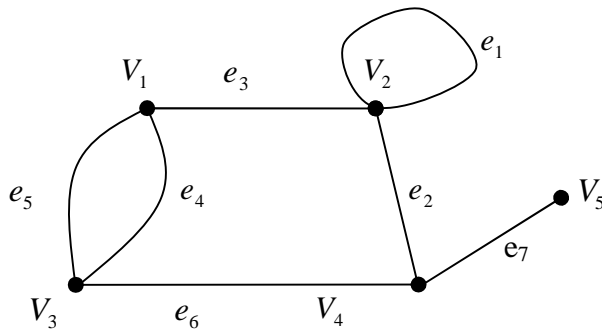
Graf didefinisikan sebagai graf $G=(V,E)$ yang terdiri dari suatu himpunan objek $V=(v_1, v_2, \dots, v_n)$ yang disebut verteks dan himpunan lain $E=(e_1, e_2, \dots, e_n)$ yang elemen-elemennya disebut edge, sedemikian sehingga tiap-tiap edge e_n diidentifikasi dengan pasangan verteks tak urut (Even, 1979). Menurut Wilson (1990), sebuah edge biasanya akan menghubungkan dua buah verteks yang berlainan, namun akan ada kasus dimana edge menghubungkan 1 verteks yang sama (v_i, v_i).

Lintasan didefinisikan sebagai deret bergantian terhitung dari verteks dan edge. Lintasan dimulai dan diakhiri dengan verteks, sehingga tiap-tiap edge berpengaruh dengan verteks yang mendahului dan mengikutinya. Pada Gambar 2, misalkan V_1 a V_2 b V_3 c V_3 d V_4 e V_2 f V_5 adalah lintasan yang ditunjukkan dengan garis tebal. Lintasan tertutup dimana tak ada verteks kecuali verteks awal dan akhir muncul lebih dari sekali disebut untai. Untai ditunjukkan pada Gambar 2, yaitu: V_2 b V_3 d V_4 e V_2 . Untai juga disebut *untai, elementary untai, circular path* dan *polygon* (Gross, 1998).

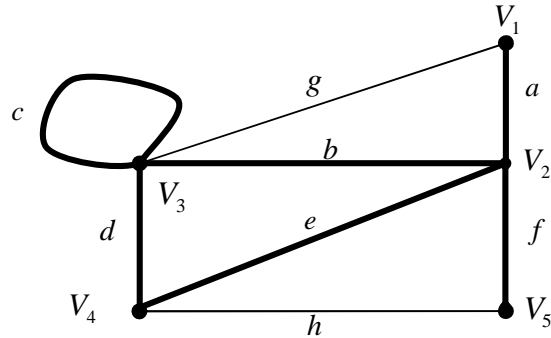
Edge dari sebuah graf mempunyai arti lebih selain sebagai penghubung. Nilai yang dibawa oleh edge dapat mempunyai banyak arti disesuaikan dengan kasus yang direpresentasikan ke graf. Untuk merepresentasikan jarak antara verteks maka pada edge diberi bobot. Suatu graf disebut graf berbobot jika tiap edge yang disebut sebagai bobot dari e , dimana e menghubungkan dua verteks i dan j (Wilson, 1990).

Graf berarah (*directed graf* atau singkatnya *digraf*) G berisi himpunan verteks-verteks $V=\{v_1, v_2, \dots\}$ dan daftar pasangan berurut $A=\{a_1, a_2, \dots\}$ dari elemen tersebut yang disebut dengan arc (Wilson, 1990). Jika v_i dan v_j

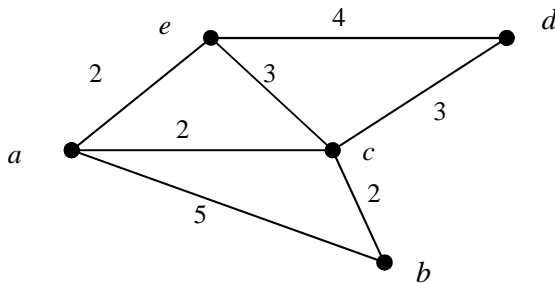
adalah verteks dari V , maka $ah(v_i, v_j)$ adalah arc h dengan bentuk $v_i v_j$ dikatakan berarah dari v_i ke v_j atau menggabungkan v_i ke v_j .



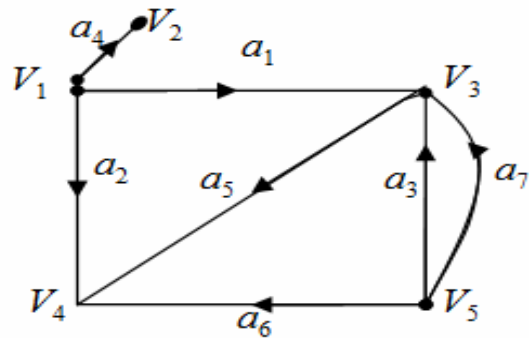
Gambar 1. Graf dengan 5 Verteks dan 7 Edge



Gambar 2. Sebuah Lintasan



Gambar 3. Graf Berbobot dengan 5 Verteks



Gambar 4. Graf Berarah dengan 5 Verteks dan 7 Arc

2.2 Metode Pencarian Lintasan Terpendek

Metode pencarian lintasan terpendek dengan Algoritma Lintasan Terpendek (*Shortest Path Algorithm*), jarak terpendek dan rute lintasan dari verteks awal ke verteks yang dituju yaitu verteks akhir dapat ditemukan dengan algoritma (Wilson, 1990):

- Langkah pertama: Beri potensial verteks awal = 0. Labelkan tiap verteks v yang dicapai langsung dari verteks awal dengan jarak dari verteks awal ke v . Pilih label terkecil dan dibuat menjadi potensial dari verteks.
- Langkah umum:
 - Pandang verteks sebagai potensial
 - Untuk tiap verteks v . Lihat verteks w yang dicapai langsung dari v dan masukkan label w , yaitu:

$$w = (\text{potensial } v) + (\text{jarak dari } v \text{ ke } w), \text{ jika } w \text{ bukan label yang terkecil.}$$
 - Jika semua verteks w telah dilabelkan pilih label terkecil didalam network yang bukan merupakan potensial.
 - Ulangi langkah umum dengan potensial baru.
 - STOP: Bila verteks akhir telah diberi tanda potensial. Ini merupakan jarak terpendek dari verteks awal ke verteks akhir.
 - Rute Lintasan Terpendek: Mulai menghitung terbalik dari verteks akhir ke verteks awal (dicapai secara backward/mundur) dan verteks vw dimasukkan bila:

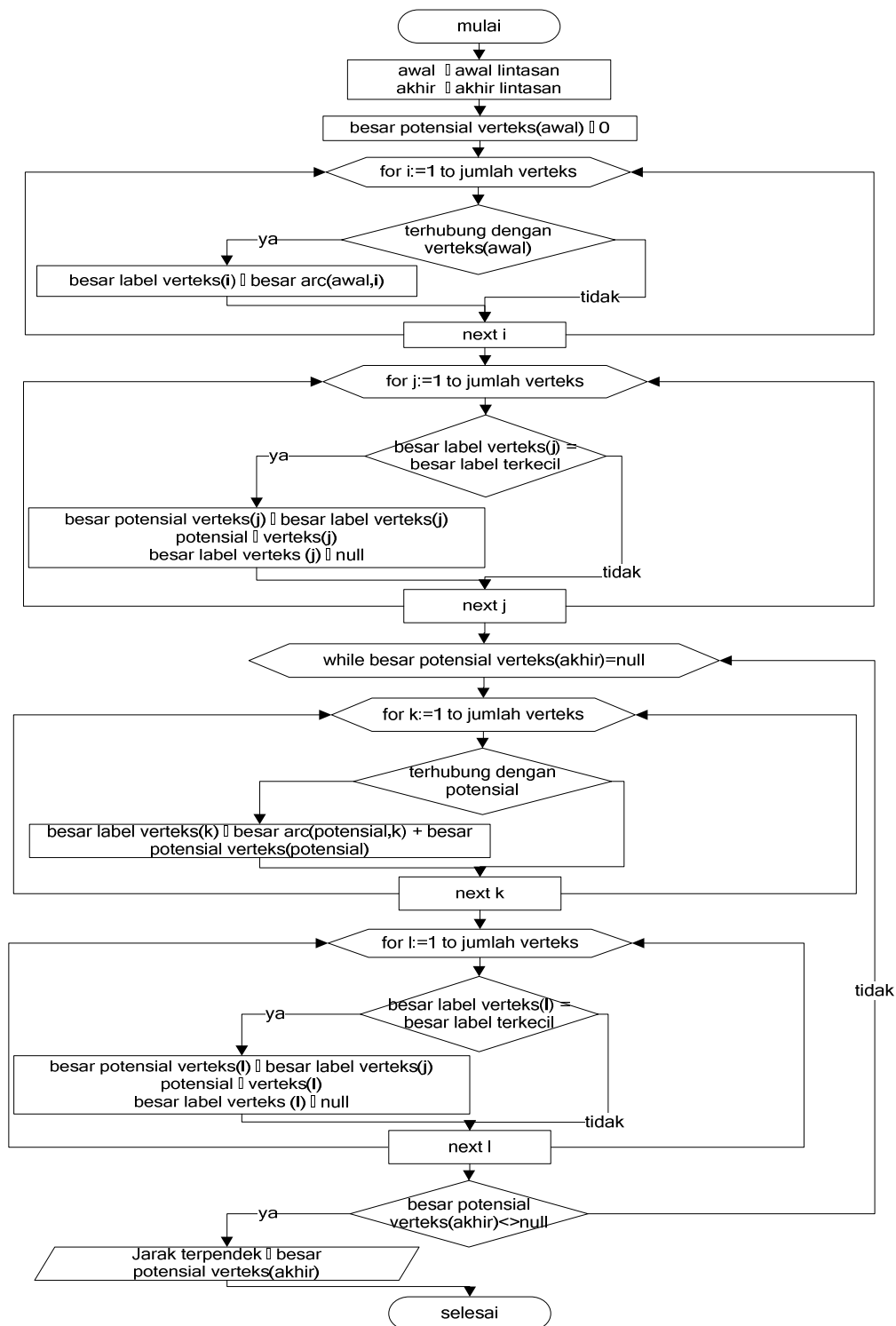
$$(\text{potensial } w) - (\text{potensial } v) = (\text{jarak dari } v \text{ ke } w)$$

3. PERANCANGAN SISTEM

Sebuah graf bersambung berarah berantai minimal harus memiliki dua buah verteks dan sebuah arc yang menghubungkannya. Sebuah verteks memiliki data-data seperti koordinat x dan koordinat y serta nama yang perlu disimpan. Penyimpanan verteks pada sistem dapat menggunakan penyimpanan larik (*array*). Penyimpanan arc hampir sama dengan verteks, sama-sama menggunakan larik (*array*). Data-data arc yang perlu disimpan adalah verteks awal, verteks akhir, bobot dan nama arc yang disimpan. Pencarian lintasan terpendek pada sebuah graf memerlukan data-data. Lintasan terpendek pada graf memiliki data-data seperti besar potensial, besar label, keterangan dan nama verteks. Penyimpanan data-data lintasan terpendek pada sistem menggunakan larik (*array*).

3.1 Algoritma Pemrograman

Langkah pertama yang harus dilakukan adalah memberikan potensial verteks awal = 0. Masing-masing verteks yang terhubung verteks awal graf diberikan label. Pilih label terkecil dan jadikan potensial berikutnya. Langkah umum yang harus dilakukan adalah memberikan label pada verteks yang terhubung dengan verteks potensial. Langkah umum dilakukan selama verteks akhir graf bukan verteks potensial. Verteks potensial merupakan verteks terpilih dengan nilai label terkecil. Gambar 5 merupakan gambar diagram alir algoritma pencarian lintasan terpendek menggunakan metode SPA.



Gambar 5. Diagram Alir Pencarian Lintasan Terpendek

3.2 Perancangan Antarmuka

Secara umum, kebutuhan terhadap antarmuka yang akan dibuat sebaiknya harus bersifat *user friendly* (ramah terhadap pengguna / mudah pengoperasiannya) dan interaktif artinya *user* sebagai pengguna terakhir dapat menggunakan sistem yang dibuat semudah mungkin sehingga *user* mengoperasikannya dengan mudah. Dengan mempertimbangkan berbagai aspek, maka dibuatlah perancangan antarmuka tampak seperti pada Gambar 6.



Gambar 6. Perancangan Antarmuka

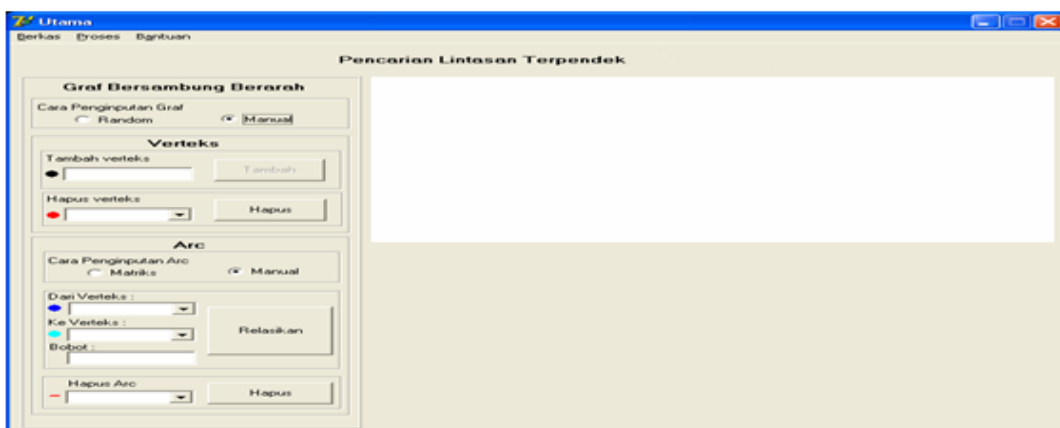
Keterangan gambar :

- Menu
 - Berkas, yang terdiri dari: buka, baru, simpan dan keluar
 - Proses Pengembangan Pencarian Lintasan Terpendek.
- Tambah dan Hapus Verteks. Terdiri dari sejumlah komponen untuk menambah dan menghapus verteks.
- Tambah dan Hapus Arc. Terdiri dari sejumlah komponen untuk menambah dan menghapus arc.
- Pilih Verteks Awal dan Akhir Graf. Terdiri dari sejumlah komponen untuk memilih verteks awal dan verteks akhir lintasan pada graf.
- Bidang Gambar Graf. Daerah ini digunakan untuk membuat gambar graf secara visual dan manual serta menampilkan gambar graf yang sudah disimpan
- Hasil Pencarian. Menampilkan jarak dan lintasan terpendek.

4. IMPLEMENTASI DAN ANALISIS SISTEM

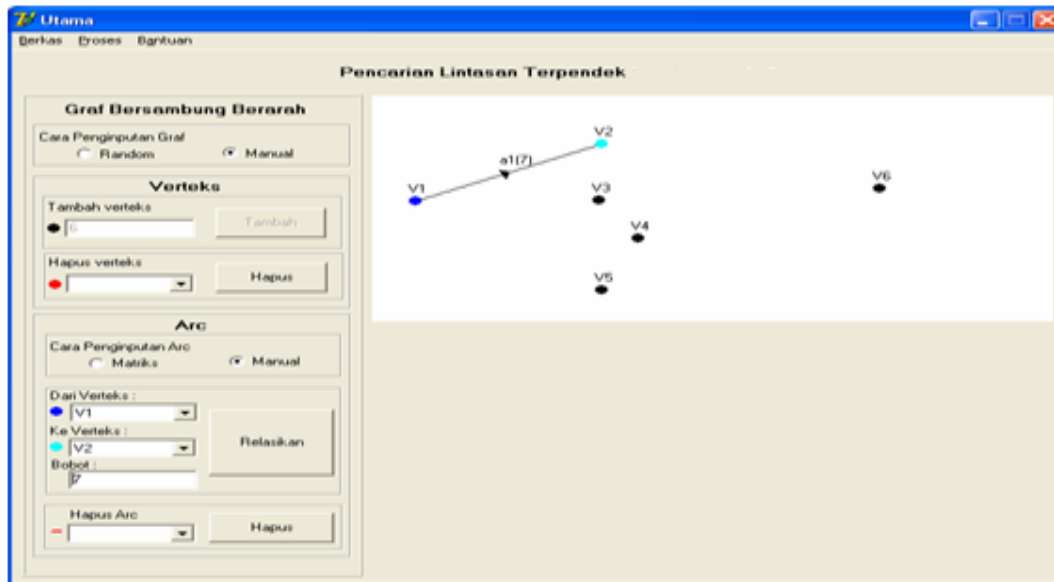
4.1 Implementasi Sistem

Bila aplikasi dijalankan, maka akan terlihat tampilan awal sistem seperti yang terlihat pada Gambar 7.



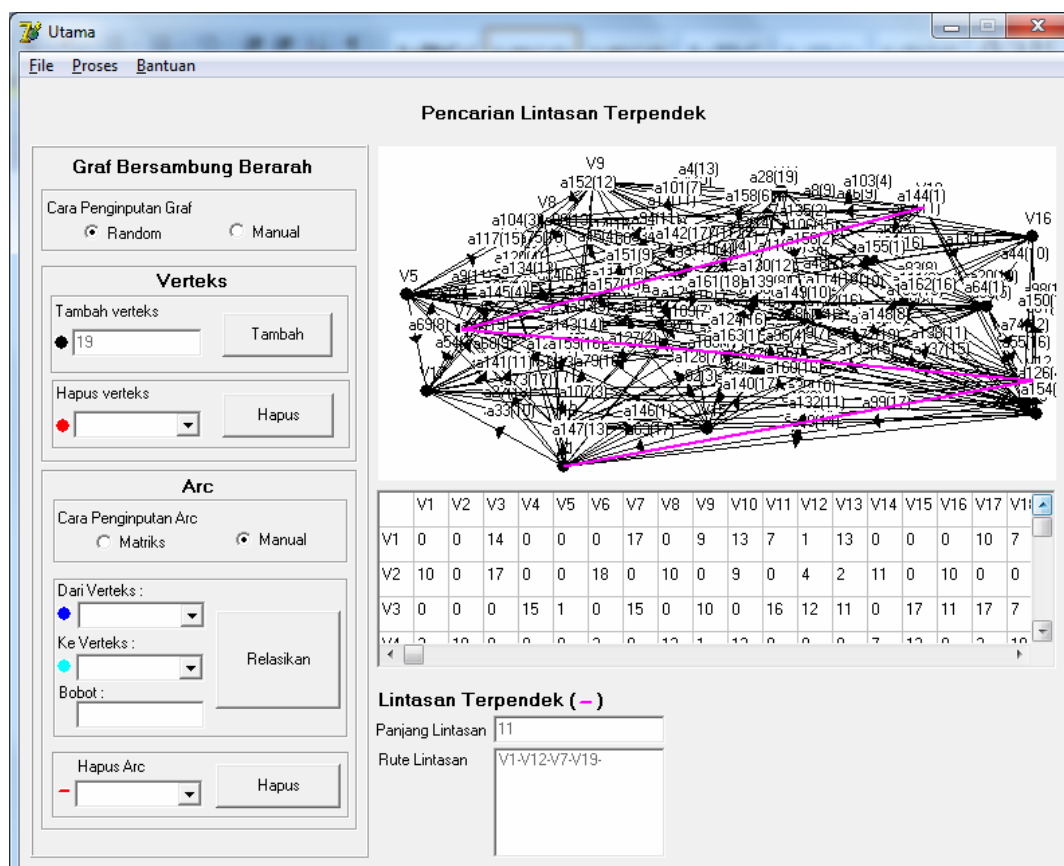
Gambar 7. Tampilan Awal Program

Proses selanjutnya adalah penginputan graf. Untuk menginput graf maka harus ditentukan terlebih dulu verteks dan arc. Aplikasi menyimpan data verteks dan arc. Tampilannya dapat dilihat pada Gambar 8.



Gambar 8. Input Graf

Proses setelah input graf adalah pencarian lintasan terpendek. Berdasar penentuan verteks awal dan akhir, aplikasi mencari lintasan terpendek dari verteks awal ke akhir. Tampilan pencarian lintasan terpendek pada graf dengan 19 verteks dan 241 arc dapat dilihat pada Modul Pencarian Lintasan Terpendek. Hasil pencarian lintasan terpendek pada graf dengan 19 verteks dan 241 arc dapat dilihat pada Gambar 9.



<pre> for o:=1 to jmlarc do for p:=1 to jmlgraf2 do begin if (q=(arrarc[o].vakhir)) and (((arrarc[o].vawal)=(arrgraf2[p].verteks2))) then begin if ((arrgraf2[p].besarpotensial2)<>"") then begin if ((arrgraf2[p].besarpotensial2)='0') then begin if (arrgraf2[p].verteks2<>combobox5.Text) then begin besarlabel:=99999999; if ((besarpotensial)>=(besarlabel)) then besarpotensial:=besarlabel; end else if (arrgraf2[p].verteks2=combobox5.Text) then </pre>	<pre> begin besarlabel:=(0+strtoint(arrarc[o].besar)); if (besarpotensial)>=(besarlabel) then besarpotensial:=besarlabel; end; end end; else begin besarlabel:=strtoint(arrgraf2[p].besarpotensial 2)+strtoint(arrarc[o].besar); if (besarpotensial)>=(besarlabel) then besarpotensial:=besarlabel; end; end; </pre>
--	--

4.2 Analisis Sistem

Utama

File Proses Bantuan

Pencarian Lintasan Terpendek

Graf Bersambung Berarah

Cara Penginputan Graf

☐ Random ☒ Manual

Verteks

Tambah verteks

5

Hapus verteks

1

Arc

Cara Penginputan Arc

☐ Matriks ☒ Manual

Dari Verteks :

1

Ke Verteks :

2

Bobot :

5

Hapus Arc

1

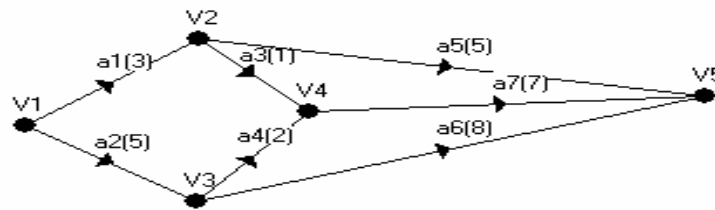
	V1	V2	V3	V4	V5
V1	0	3	5	0	0
V2	0	0	0	1	5
V3	0	0	0	2	8
V4	0	0	0	0	7

Lintasan Terpendek (-)

Panjang Lintasan 8

Rute Lintasan V1-V2-V5-

D-285



Gambar 11. Graf Bersambung Berarah Tanpa Untai

Dari Gambar 11, metode pencarian lintasan terpendek dari verteks V1 ke verteks V5 dengan beberapa iterasi dapat ditabulasikan sama seperti pada pencarian lintasan terpendek menggunakan pengembangan metode SPA. Metode SPA dikembangkan untuk memecahkan permasalahan metode SPA dalam pencarian lintasan terpendek pada graf bersambung berarah beruntai. Metode pencarian lintasan terpendek dari verteks V1 ke verteks V5 dengan beberapa iterasi dapat ditabulasikan sebagai berikut:

- Iterasi 0

Tabel 1.

Pencarian lintasan terpendek
menggunakan metode SPA (iterasi 0)

Verteks	Besar Potensial	Besar Label	Keterangan
V1	0	-	0
V2	3	-	1
V3	-	5	1
V4	-	-	1
V5	-	-	1

Beri nilai besar potensial V1 = 0. Beri nilai 1 pada semua keterangan verteks. Labelkan dengan nilai yang terkecil tiap verteks v bukan potensial yang dicapai dari verteks dengan besar potensial dan keterangan = 1 (verteks V1) yaitu verteks V2 dan verteks V3 dengan jarak dari verteks V1. Keterangan verteks V1 = 0. Pilih verteks berlabel terkecil dari semua label yang ada (verteks V2), jadikan besar labelnya sebagai nilai besar potensial.

- Iterasi 1

Tabel 2.

Pencarian lintasan terpendek
menggunakan metode SPA (iterasi 1)

Verteks	Besar Potensial	Besar Label	Keterangan
V1	0	-	0
V2	3	-	0
V3	-	5	1
V4	4	-	1
V5	-	8	1

Labelkan dengan nilai yang terkecil tiap verteks v bukan potensial yang dicapai dari verteks dengan besar potensial dan keterangan = 1 (verteks V2) yaitu verteks V4 dan verteks V5 dengan jarak dari verteks V2, lalu keterangan verteks V2 = 0. Pilih label terkecil dari semua label yang ada (verteks V4), jadikan besar labelnya sebagai nilai besar potensial.

- Iterasi 2

Tabel 3.

Pencarian lintasan terpendek
menggunakan metode SPA (iterasi 2)

Verteks	Besar Potensial	Besar Label	Keterangan
V1	0	-	0
V2	3	-	0
V3	5	-	1
V4	4	-	0
V5	-	8	1

Labelkan dengan nilai yang terkecil tiap verteks v bukan potensial yang dicapai dari verteks dengan besar potensial dan keterangan = 1 (verteks V4) yaitu verteks V5, lalu keterangan verteks V4 = 0. Pilih label terkecil dari semua label yang ada (verteks V3), jadikan besar labelnya sebagai nilai besar potensial.

- Iterasi 3

Tabel 4.
Pencarian lintasan terpendek
menggunakan metode SPA (iterasi 3)

Verteks	Besar Potensial	Besar Label	Keterangan
V1	0	-	0
V2	3	-	0
V3	5	-	0
V4	4	-	0
V5	8	-	1

Labelkan dengan nilai yang terkecil tiap verteks v bukan potensial yang dicapai dari verteks dengan besar potensial dan keterangan = 1 (verteks V3) yaitu verteks V5, lalu keterangan verteks potensial=0. Pilih label terkecil dari semua label yang ada (verteks V5), jadikan besar labelnya sebagai nilai besar potensial. Iterasi dihentikan, verteks V5 telah diberi tanda potensial. Jarak terpendek dari verteks V1 ke verteks V5 = 8. Lintasan terpendek dicapai secara mundur (*backward*).

- Besar potensial V5 – besar potensial V2 = jarak dari V2 ke V5 ($8 - 3 = 5$). Benar
Lintasan terpendek: -V2-V5
- Besar potensial V2 – besar potensial V1 = jarak dari V1 ke V2 ($3 - 0 = 3$). Benar
Lintasan terpendek: V1-V2-V5

Pencarian lintasan dihentikan, lintasan terpendek dari verteks V1 ke verteks V5 adalah V1-V2-V5

5. KESIMPULAN DAN SARAN

Bedasar hasil analisis penelitian dan pengujian pada aplikasi ini dapat disimpulkan beberapa hal sebagai berikut:

- Pengembangan SPA berhasil mencari jalur terpendek pada graf bersambung berarah berantai.
- Aplikasi komputer pada pengembangan SPA berhasil mencari jalur terpendek secara cepat dan tepat berbanding pencarian secara manual.

Pada penelitian ini masih banyak terdapat kekurangan yang perlu diteliti lebih lanjut sebagai bahan untuk pengembangan sistem. Beberapa saran yang dapat diberikan guna pengembangan penelitian selanjutnya:

- Penambahan arc dikembangkan dengan cara visual.
- Penambahan jumlah maksimum penginputan verteks yang diijinkan.
- Penambahan besar bobot arc yang diijinkan.

DAFTAR PUSTAKA

Buckley, Fred. 2002. *A Friendly Introduction To Graph Theory*. New Jersey: Pearson Education, Inc.
Even, Shimon. 1979. *Graph Algorithms*. United State of America : Computer Science Press.
Gross, Jonathan. 1998. *Graph Theory and Its Applications*. Florida: CRC Press.
Wilson, Robin J. 1990. *Graphs: An Introductory Approach*. Canada : John Wiley & Sons.